Title page with author details

Yanru Zhong

School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, PR China

Chaohao Jiang[1]

School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, PR China

Yuchu Qin

School of Computing and Engineering, University of Huddersfield, Huddersfield, HD1 3DH, UK

Guoyu Yang

School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, PR China

Meifa Huang

School of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, Guilin 541004, PR China

Xiaonan Luo

School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, PR China

[1]Corresponding author.
E-mail address: chaohaojiang@outlook.com(C.Jiang).

Automatically generating assembly sequences with an ontology-based approach

Abstract

Purpose-The purpose of this paper is to present and develop an ontology-based approach for automatic generation of assembly sequences.

Design/methodology/approach-In this approach, an assembly sequence planning ontology is constructed to represent the structure and interrelationship of product geometry information and assembly process information. In the constructed ontology, a number of reasoning rules are defined to describe assembly knowledge and experience. Based on the ontology with reasoning rules, an algorithm for automatically generating assembly sequences is designed and implemented.

Findings-The effectiveness of this approach is verified via applying it to generate the assembly sequences of a gear reducer.

Originality/value-The main contribution of the paper is presenting and developing an ontology-based approach for automatically generating assembly sequences. This approach can provide a feasible solution for the issue that mathematics based assembly sequence generation approaches have great difficulty in explicitly representing assembly experience and knowledge.

Papertype Research Paper

Keywords: Assembly sequence, Automatic generation, Assembly model, Ontology, Semantics, Rule-based reasoning.

1. Introduction

Assembly sequence planning (ASP) plays an important role in product development. According to statistics, assembly activities account for about 50% of product manufacturing resource consumption (Ou and Xu, 2013). Since 1980, computers have been used as auxiliary tools to support the study of ASP to improve manual approaches. Researchers have developed a variety of models and solutions (Wang et al., 2005a; Xu et al., 2011).

Although these approaches and techniques have been studied extensively and intensively, representation of assembly experience and knowledge still requires further in-depth integration with the existing calculation approaches. ASP is a process based on experience of engineers, assembly knowledge, and geometric information of the product. However, the geometric constraints and assembly features of parts are used as the main calculation basis for assembly sequences. Without the support of experience and knowledge, the inferred sequences may not be feasible in engineering. Algorithms can be designed to be more complex to consider the requirements of assembly knowledge. But this will bring a large number of redundant calculations.

Due to the requirements of assembly knowledge representation in ASP, ontology technology was introduced. Ontology is an explicit formal specification of the shared conceptual model (Gruber, 1993). It can integrate different knowledge from different sources, realize knowledge sharing, describe different granular concepts, and support logical reasoning (Fensel, 2003). Ontology has advantages in terms of semantic expression, intelligent reasoning, and knowledge reuse (Staab and Studer, 2010).

Based on ontology technology, an automatic assembly sequence generation approach is proposed in this paper. In this approach, ontology and rules are firstly used to describe assembly knowledge and experience. Assembly sequences are divided automatically by reasoning, which include the fixed sequence of special parts, the sequence of ordinary parts, and the sequence of connectors. Then, an algorithm based on the existing methods is designed and improved to obtain the sequences of parts.

The rest of the paper is organized as follows. Section 2 provides an overview of related work. An assembly model is established in Section 3. Section 4 explains the details of the proposed ontology-based approach. In Section 5, generation of the assembly sequences of a reducer is used as an example to verify the effectiveness of the approach. Section 6 ends the paper with conclusions.

2. Related work

The efficiency of assembly sequence generation is mainly affected by the way for modeling assembly information. During the past few decades, a number of modeling methods for function, product structure, and assembly semantics have been presented. Umeda et al. (1996) designed a Function-Behavior-State modeling method. Zhang et al. (2005) presented a graph and matrix representation scheme for functional design. Chen et al. (2012) proposed a multi-level assembly model based on a top-down manner. Zhu et al. (2012) proposed an ontological reasoning mechanism to infer implicit information in product model. Kim et al. (2006) made detailed classification work on assembly semantics and described their model using ontologies.

The goal of ASP is to automatically generate assembly sequences that meet actual assembly requirements using assembly knowledge and constraints. The mainstream methods for ASP include precedence constraint relation based method, disassembly based method, knowledge based method, heuristic algorithm based method, etc. Bourjault (1984) proposed the concept of assembly sequence priority relation, in which prioritized relations are obtained by answering a series of design questions. Baldwin et al. (1991) built an integrated set of user interactive computer programs. The program uses disassembly analysis to generate sequences. Yin et al. (2003) proposed a connector-based method for ASP. Lv et al. (2010) used hybrid algorithms to optimize product assembly sequences. Lu and Yang (2016) proposed a method for ASP and assembly line balance using ant colony algorithm. Zhao and Li (2009) proposed a formalized reasoning method based on polychromatic set theory. These methods generally rely on humancomputer interaction to obtain assembly knowledge and screening sequences. It is difficult for heuristic-based algorithms to comprehensively embed assembly knowledge and experience.

To reduce computational complexity and automatically acquire assembly knowledge of products, many researchers use advanced expert knowledge or experience to plan sequences. Swaminathan et al. (1998) summarized commonly used assembly structures and assembly sequences. Dong et al. (2007) provided a way to consider geometric information and non-geometric knowledge. These methods are still limited in terms of knowledge reasoning and sharing. Therefore, ontology was introduced into ASP because of its advantages. Meng et al. (2016) designed assembly ontology for ASP, and obtained the priority relation through rule inference. Qiao et al. (2018) proposed a geometry-enhanced ontology model and reasoning framework. Jiang et al. (2018) constructed ontology to describe product information. The disassembly tool and sequence are obtained by decision support methods of design and reasoning. Cheng et al. (2012) presented an ontology-driven case-based reasoning method. Chen et al. (2016b) employed ontology concept on the disassembly decision making process using case-based reasoning. Barbau et al. (2012) proposed a STEP-based ontology to translate the product geometric information. Cochrane et al. (2009) proposed a Process Specification Language based ontology to represent process planning knowledge. Gruhier et al. (2015) introduced the development of a formal ontology in the context of integrated assembly design and ASP. Zhu and Roy (2018) developed a disassembly information model based on various information aspects in demolition planning area.

As can be summarized from the above literature review, assembly knowledge and experience need to be further integrated into the ASP methods based on geometric information calculations to reduce the amount of computation and produce sequences that conform to assembly habits. Knowledge-based reasoning method is limited by computational efficiency. Similarly, methods based on geometric information calculations are also limited by assembly knowledge.

Therefore, this paper uses a combination of knowledge representation and other computational methods to generate sequences.

3. Assembly modeling

3.1. Notations

The notations used in the paper are defined as follows:

$S_{SAP,i}$	= a subassembly planning set, $i \in [1, n_1]$;
$A_{S,n}$	= a subassembly, $n \in [1, n_2]$;
P_i	=a part, $i \in [1, n_3]$
$P_{C,i}$	= a connector, $i \in [1, n_4]$;
$G_{AP}(x, y)$	= a group of parts consisting of $A_{S,n}, P_i$ or $P_{C,i}$;
PR, S_{PR}, E_{PR}	= respectively the positioning relation, the set of positioning relation and the logic equation of positioning relation;
IR, S_{IR}, E_{IR}	= respectively the interference relation, the set of interference relation and the logic equation of interference relation;
$S_{IR}i$	= the interference relation set in the i direction, $S_{IR}i \in S_{IR}$ and $i \in \pm X, \pm Y, \pm Z$;
SR, S_{SR}, E_{SR}	= respectively the support relation, the set of support relation and the logic equation of support relation;
CR, S_{CR}	= the connection relation and the set of connection relation;
S_{CRA}	= the connection relation set in the case where the installation sequence is part-part-connector, $S_{CRA} \in S_{CR}$;
S_{CRB}	= the connection relation set in the case where the installation sequence is part-connector-part, $S_{CRB} \in S_{CR}$;
E_{CRA}	= the logic equation of connection relation in the case where the installation sequence is part-part-connector;
E_{CRB}	= the logic equation of connection relation in the case where the installation sequence is part-connector-part;
SP	= special installation regulations;
AR	= adjacent relation;
S_{after}	= A set of parts that are required to be installed after the part P_i is installed.
Φ	= an empty set.

3.2. Assembly model

The assembly model serves as an integration center for product assembly information. Its constituent elements, hierarchical structure, and information transmission have a guiding role in the approach of ASP. The framework of the model is shown in Fig. 1. As can be seen from the figure, the model uses a top-down approach, which starts with assembly. Through the analysis of subassemblies and assembly constraints, assembly information and experience are attributed to the part level. In Fig. 1, S_T, S_A, and S_C are respectively part types, attributes, and assembly constraints.

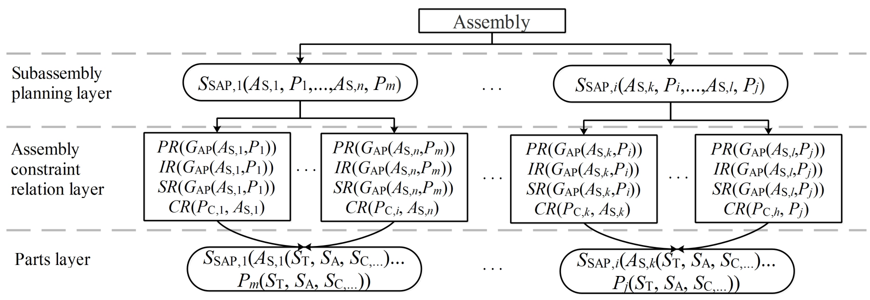


Figure 1: Framework of the assembly model.

The subassembly planning layer is the first layer of the assembly model. Its main function is to divide the assembly into several subassemblies according to the requirements of functions, structures, and assembly processes. This layer is designed to deal with a large number of parts in a stratified and step-by-step manner (Chen et al., 2016a).

The assembly constraint layer is the second layer of the model. Its main function is to represent the constraint relations between parts.

(1) Positioning relation. The process of determining the position of a part can be seen as a process in which the movement and rotation of the part are limited in three-dimensional space (Kim et al., 2005). When the movement and rotation of the part are limited by the contact surface (Gao et al., 2004), the position of the part in the assembly is determined.

(2) Interference relation. An effective assembly sequence must satisfy geometric constraints to ensure that there is a viable mounting channel during the assembly process. The geometric constraints can be fully described by interference relations (Zhang et al., 2016).

(3) Support relation. For virtual assembly in computer environment, the three-dimensional space does not need to consider gravity factor. However, in actual assembly process, parts need the support of fixtures and other parts to ensure stable installation under the influence of gravity.

(4) Connection relation. The fixation and connection of the connectors are the main means to ensure the stable installation of the assembly(Yu et al., 2010).

The part layer is the third layer of the model. Its main function is to inherit the assembly constraint information passed by the upper layer, and describe the geometric information, attribute information, and structure information of the part. The definition of this layer is as follow:

The part layer is a triple (S_T, S_B, S_P), where $S_T = \{OP, SA, C\}$ indicates the type of parts (where OP indicates ordinary part, SA indicates subassembly, and C indicates connector), $S_B = \{N, ID, V, M, PN, T\}$ represents the basic information of a part (where N indicates the name of the part, ID indicates the number of the part, V indicates the volume of the part, M indicates the material of the part, PN indicates product model numbers, and T indicates processing tool), and $S_P = \{S_{AC}, BBox, 3DM, GA\}$ denotes the assembly constraints and design document information (where $S_{AC} = \{S_{PR}, S_{IR}, S_{SR}, S_{CR}\}$ denotes a set of assembly constraints, $BBox$ represents bounding box, $3DM$ represents $3D$ model annotation, and GA represents graphical annotation).

Assume the logic value of a part that is not mounted is 0, and the logic value of the mounted part is 1. When the logic equation $E_{PR} = \bigvee P_i \bigvee A_{S,j} = 1$, then it meets installation conditions, otherwise it does not meet installation conditions. P_i and $A_{S,j}$ belong to S_{PR}. Similarly, when $E_{IR} = \bigwedge S_{IR}i = 0$, then it meets installation conditions, where $S_{IR}i = \bigvee P_i \bigvee A_{S,j} \bigvee P_{C,i}$, $P_{C,i}$, P_i and $A_{S,j}$ belong to $S_{IR}i$. If there is an empty set of $S_{IR}i$, then $E_{IR} = 0$. Simultaneously, when $S_{after} \neq \Phi$, then $E_{IR} = (\bigwedge S_{IR}i) \bigvee P_i \bigvee A_{S,j}$. P_i and $A_{S,j}$ belong to S_{after}. When $E_{SR} = \bigvee P_i \bigvee A_{S,j} = 1$, then there are installation conditions. P_i and $A_{S,j}$ belong to S_{SR}. If S_{SR} is an empty set, then $E_{SR} = 1$. When $E_{CRA} = \bigwedge P_i \bigwedge A_{S,j} \bigwedge P_{C,i} = 1$, then there are installation conditions. $P_{C,i}$, P_i and $A_{S,j}$ belong to S_{CRA}. When $E_{CRB} = \bigvee P_i \bigvee A_{S,j} \bigvee P_{C,i} = 1$, then there are installation conditions. $P_{C,i}$, P_i and $A_{S,j}$ belong to S_{CRB}.

4. Ontology-based approach

4.1. Overview of the approach

The ontology-based approach centers on the semantic model of assembly experience and knowledge (Jiang et al., 2018; Chen et al., 2016b). Unlike pure mathematics methods, ontology-based hybrid approach has the ability to explicitly represent data semantics, thus it can effectively describe assembly experience. An overview of this approach is depicted in Fig. 2. The specific modules of the approach are as follows:

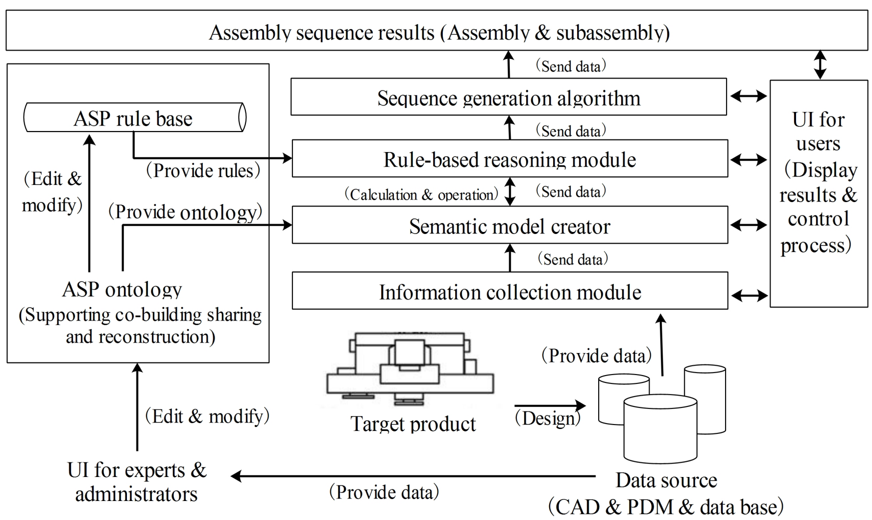


Figure 2: Overview of the ontology-based automatic assembly sequence generation approach.

(1) ASP ontology. The ontology defines the assembly knowledge formally. All the assembly knowledge in the sequence generation approach is expressed and stored by the ontology.

(2) Assembly rule base. This module stores assembly rules created by experts. The rules are created based on assembly experience and knowledge.

(3) Assembly information integration module. This module analyzes and extracts assembly constraints, hierarchies, and part properties from CAD/PDM

files and databases through the software's functional modules and expert methods.

(4) Assembly semantics generation module. The module needs the ASP ontology to provide the semantic basis, map the assembly information to the semantic model represented by the ontology, and instantiate the ontology.

(5) Assembly rule inference module. The module needs the support of the ontology and rules. It obtains new assembly information through inference steps.

(6) Automatic sequence generation module. The sequence generation algorithm is designed and extended to produce the final result. The algorithm can be traversal algorithm, genetic algorithm, ant colony algorithm, etc.

4.2. Construction of the ontology

This paper adopts the seven-step method to construct the ontology (Fernández-López and Gómez-Pérez, 2002) and uses OWL language and Prot*égé* editor to describe and edit the ontology (McGuinness and Harmelen; OWL Working Group; Stanford Center for Biomedical Informatics Research). First, define classes and hierarchies. According to the assembly model, the term of unary relations is defined as a class. The classes in the ontology and their hierarchical relations are shown in Fig. 3(a). Second, define properties. Terms that represent binary relations can be defined as properties. The defined properties are shown in Fig. 3(b). Their usages are explained as follows:

(1) Data Properties. They are used to represent various data of the part.

(2) Object Property 1-4 describes the reference part of the part group and the hierarchical relation.

(3) Object Property 5-7 indicates the assembly constraint relations among parts. The assembly relations include the adjacency relation, interference relation, and connection relation.

(4) Object Property 8-13 involves part, connector, and special installation requirements. They describe installation type, tool and method.

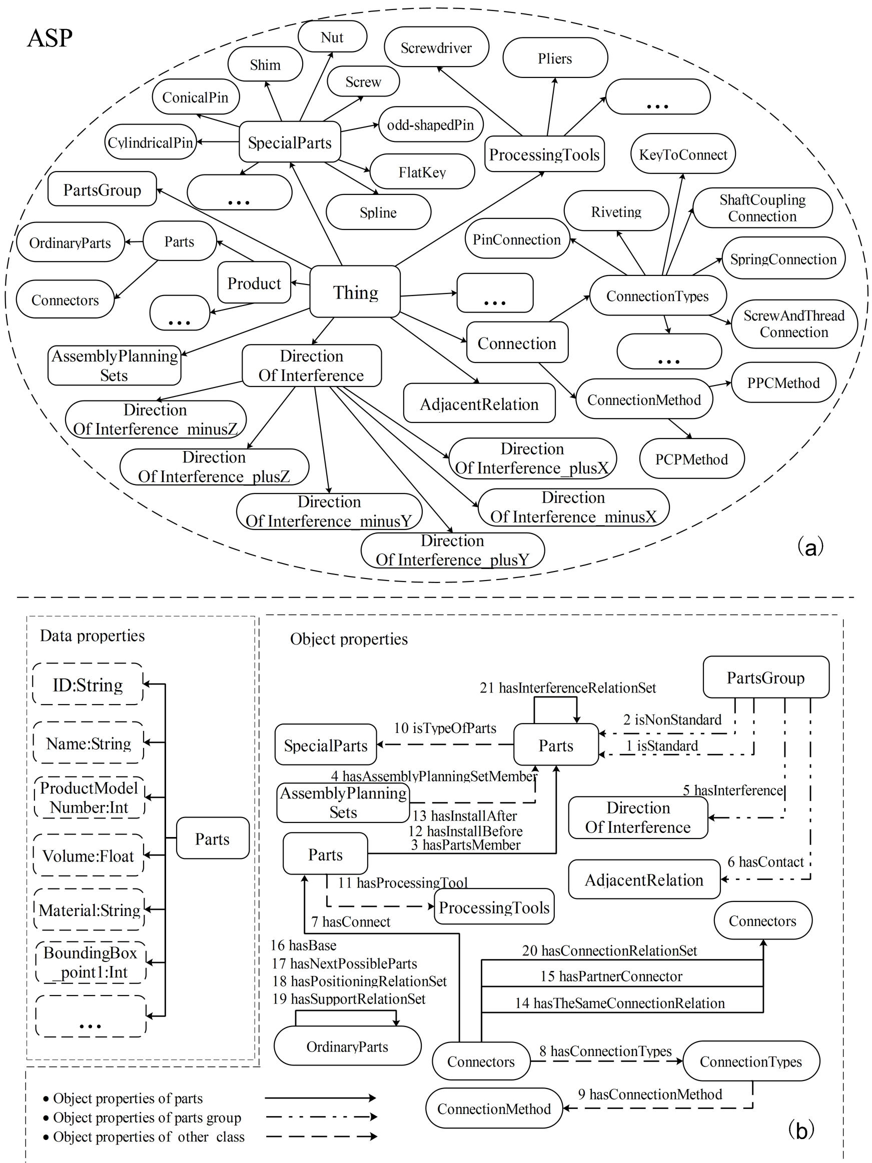


Figure 3: Schematic representation of the constructed ASP ontology.

(5) Object Property 14-15 involves the connectors and connector groups. Property 14, 15 represents the relation between the connectors.

(6) Object Property 16-21 describes the result set of the assembly constraints of the parts, which include interference, positioning, support, etc.

4.3. Assembly knowledge representation

Assembly constraint relation representation. Assembly constraints are used primarily for the calculation of ordinary part sequences in this paper, including positioning relations, interference relations, and support relations. According to the assembly ontology, assembly constraints are described using Semantic Web Rule language (SWRL) (Motik et al., 2005) and shown in the Table 1 (see S01-01 to S03-02). For example, interference relation is obtained by the following rule:

Rule 1: The part group consists of a reference part A and a non-reference part B. If A and B have an interference relation in the i direction, then the mounting path of A is blocked by B in the i direction. Similarly, the mounting path of B is blocked by A in the reverse direction j of i (if i is the $+X$, then j is the $-X$).

The SWRL rules for expressing Rule 1 are shown in Table 1 (see S02-01 to S02-06). According to the syntax and usage of SWRL (Horrocks et al.), the statement is divided into two parts: condition and conclusion. And the statement mainly involves the representation of the unary and binary relations. E.g. PartsGroup($?x$) represents all instances in the concept of parts group, where $?x$ represents one of these instances. isStandard ($?x, ?a$) indicates that the part is the reference of the part group, where $?a$ represents the variable under the concept of part.

Connection relation representation. This paper obtains the set of connection relations for different installation methods based on different types of connectors. The SWRL rules for connection relation are shown in Table 1 (see S04-01 to S04-03). The connector may affect the calculation of the constraint relation of other ordinary parts, such as the keys. However, the reasoning results

Table 1: Assembly information generation rules defined in the constructed ASP ontology.

ID	SWRL/SQWRL
S01-01	$PartsGroup(?x) \wedge AdjacentRelation(?y) \wedge hasContact(?x,?y) \wedge isStandard(?x,?c) \wedge isNonStandard(?x,?d) \wedge OrdinaryParts(?c) \wedge OrdinaryParts(?d)->hasPositioningRelationSet(?c,?d) \wedge hasPositioningRelationSet(?d,?c) \wedge hasNextPossibleParts(?c,?d) \wedge hasNextPossibleParts(?d,?c)$
S02-01	$PartsGroup(?x) \wedge DirectionOfInterference\_plusX(?y) \wedge hasInterference(?x,?y) \wedge isStandard(?x,?a) \wedge isNonStandard(?x,?b)->hasInterferenceRelationSet\_plusX(?a,?b) \wedge hasInterferenceRelationSet\_minusX(?b,?a)$
S02-02	$PartsGroup(?x) \wedge DirectionOfInterference\_minusX(?y) \wedge hasInterference(?x,?y) \wedge isStandard(?x,?a) \wedge isNonStandard(?x,?b)->hasInterferenceRelationSet\_minusX(?a,?b) \wedge hasInterferenceRelationSet\_plusX(?b,?a)$
S02-03	$PartsGroup(?x) \wedge DirectionOfInterference\_plusY(?y) \wedge hasInterference(?x,?y) \wedge isStandard(?x,?a) \wedge isNonStandard(?x,?b)->hasInterferenceRelationSet\_plusY(?a,?b) \wedge hasInterferenceRelationSet\_minusY(?b,?a)$
S02-04	$PartsGroup(?x) \wedge DirectionOfInterference\_minusY(?y) \wedge hasInterference(?x,?y) \wedge isStandard(?x,?a) \wedge isNonStandard(?x,?b)->hasInterferenceRelationSet\_minusY(?a,?b) \wedge hasInterferenceRelationSet\_plusY(?b,?a)$
S02-05	$PartsGroup(?x) \wedge DirectionOfInterference\_plusZ(?y) \wedge hasInterference(?x,?y) \wedge isStandard(?x,?a) \wedge isNonStandard(?x,?b)->hasInterferenceRelationSet\_plusZ(?a,?b) \wedge hasInterferenceRelationSet\_minusZ(?b,?a)$
S02-06	$PartsGroup(?x) \wedge DirectionOfInterference\_minusZ(?y) \wedge hasInterference(?x,?y) \wedge isStandard(?x,?a) \wedge isNonStandard(?x,?b)->hasInterferenceRelationSet\_minusZ(?a,?b) \wedge hasInterferenceRelationSet\_plusZ(?b,?a)$
S03-01	$PartsGroup(?x) \wedge DirectionOfInterference\_minusZ(?y) \wedge hasInterference(?x,?y) \wedge AdjacentRelation(?z) \wedge hasContact(?x,?z) \wedge isStandard(?x,?a) \wedge isNonStandard(?x,?b) \wedge OrdinaryParts(?a) \wedge OrdinaryParts(?b)->hasSupportRelationSet(?a,?b)$
S03-02	$PartsGroup(?x) \wedge DirectionOfInterference\_plusZ(?y) \wedge hasInterference(?x,?y) \wedge AdjacentRelation(?z) \wedge hasContact(?x,?z) \wedge isStandard(?x,?a) \wedge isNonStandard(?x,?b) \wedge OrdinaryParts(?a) \wedge OrdinaryParts(?b)->hasSupportRelationSet(?b,?a)$
S04-01	$Connectors(?x) \wedge hasConnect(?x,?y) \wedge hasConnectionTypes(?x,?a) \wedge PCPMethod(?z) \wedge hasConnectionMethod(?a,?z)->hasConnectionSet\_PCP(?x,?y)$
S04-02	$Connectors(?x) \wedge hasConnect(?x,?y) \wedge hasConnectionTypes(?x,?a) \wedge PPCMethod(?z) \wedge hasConnectionMethod(?a,?z)->hasConnectionSet\_PPC(?x,?y)$
S04-03	$hasAssemblyPlanningSetMember(?o,?x) \wedge hasAssemblyPlanningSetMember(?o,?y) \wedge hasTheSameConnectionRelation(?x,?y) \wedge ProductModelNumber(?x,?a) \wedge ProductModelNumber(?y,?b) \wedge swrlb:equal(?a,?b)->hasPartnerConnector(?x,?y) \wedge hasPartnerConnector(?y,?x)$
S05-01	$Parts(?x) \wedge Parts(?y) \wedge hasContact(?x,?y) \wedge ShaftWithGroove(?a) \wedge Key(?b) \wedge isTypeOfParts(?x,?a) \wedge isTypeOfParts(?y,?b) ->hasInstallBefore(?y,?x) \wedge hasInstallAfter(?x,?y)$
S05-02	$Parts(?x) \wedge Parts(?y) \wedge hasContact(?x,?y) \wedge Gear(?a) \wedge Key(?b) \wedge isTypeOfParts(?x,?a) \wedge isTypeOfParts(?y,?b)->hasInstallBefore(?x,?y) \wedge hasInstallAfter(?y,?x)$
S05-03	$Parts(?x) \wedge Parts(?y) \wedge hasContact(?x,?y) \wedge Gear(?a) \wedge ShaftWithGroove(?b) \wedge isTypeOfParts(?x,?a) \wedge isTypeOfParts(?y,?b) ->hasInstallBefore(?x,?y) \wedge hasInstallAfter(?y,?x)$

describe a simple connection relation. It is difficult to find a suitable installation location in an indeterminate sequence by rule reasoning. Therefore, the paper designs the sequence determination process of the connector, as shown in the Fig. 4. The judgment basis is determined according to the installation method. When the connector belongs to only one installation method, E_{CRA} or E_{CRB} is used. When the connector group is a hybrid installation, the connectors with the installation method part-part-connector are prioritized, such as a combination of screws, nuts and shims.

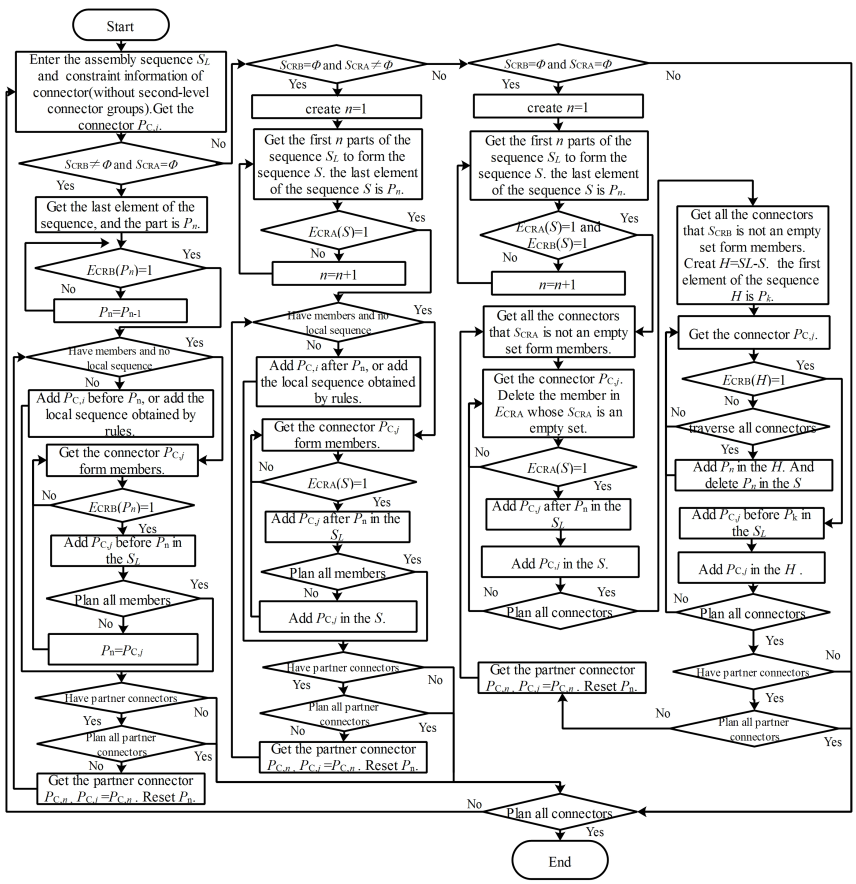


Figure 4: Process of connector sequence calculation.

Special sequence representation. During the installation of products, there are a large number of special sequences. These sequences are fixed due to the requirements of the assembly process and long-term accumulated experience. These special parts are calculated as a whole externally or controlled through the interference logic equation, and the sequence of the parts is obtained by the following rule:

Rule 2: Part A and Part B. If A and B belong to some special part, then A can be installed before B. Similarly, A can be installed after B.

The SWRL rules for expressing Rule 2 are shown in Table 1 (see S05-01 to S05-03). According to the requirements of the example, the rules for thread connection are written.

The intermediate values and results in the reasoning process require other operations to assist the inference engine. Therefore, the assembly sequence inference process is shown in Fig. 5. First, knowledge and rules are populated into the inference engine. Then, according to the steps of design, operations such as inference, creation, and update are performed. Finally, the inference results are extracted for the planning algorithm.

4.4. Automatic generation algorithm

Assembly sequences require a comprehensive consideration of assembly information. The advantage of ontology lies in the representation and reasoning of knowledge. Therefore, it is necessary to use algorithms to support the rapid generation of sequences and to process a large number of computations of complex products. The process of assembly sequence generation is gradually decomposed. The planning problem can be seen as a series of sequential assembly operations that link parts together. Then an assembly operation is defined as follow:

Assembly operation is a quad consisting of assembly basic elements, i.e. $AO = (NP, SubA, AF, AC)$, where $NP = \{NP_i \mid i \in [1, n]\}$ denotes a set of parts that are not installed, $SubA = \{SubA_{,i} \mid i \in [1, m]\}$ denotes a set of parts that have been installed, $AF = \{True, False\}$ denotes the mountability of parts, and AC denotes the cost of installing a part. When the part NP_i

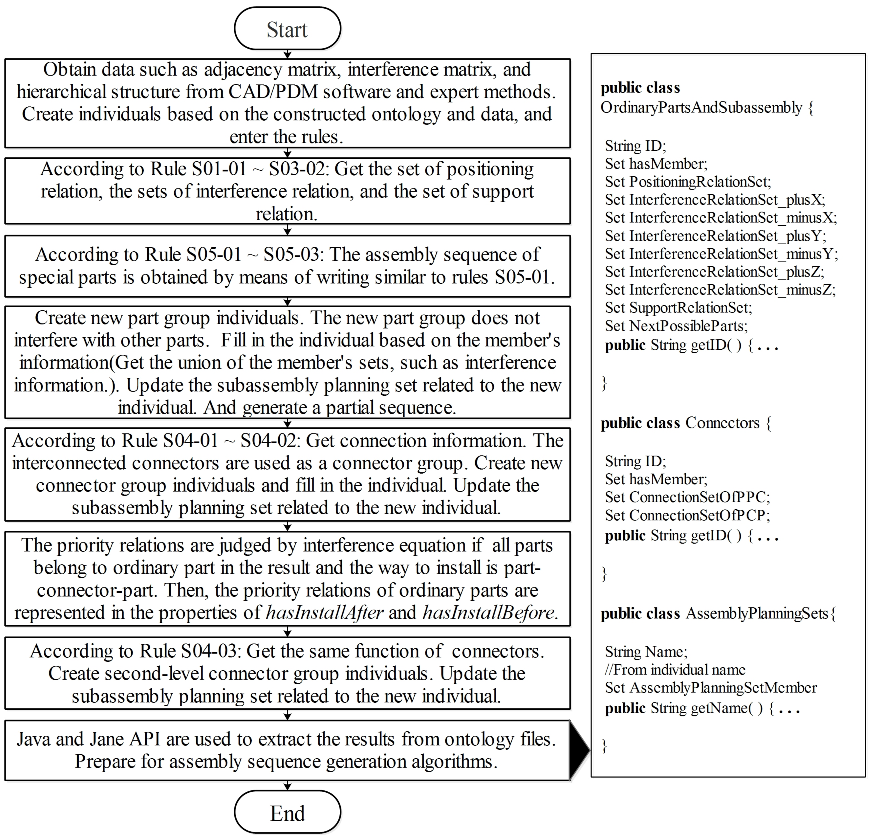


Figure 5: Process of assembly sequence inference.

has mountability, the part is needed to satisfy $E_{IR}(SubA)$. At the same time, it is required that the part NP_i does not hinder the installation of subsequent parts. That is, when $SubA^* = SubA \bigcup \{NP_i\}$ and $NP^* = NP - \{NP_i\}$, all parts $NP_j \in NP^*$ satisfy $E_{IR}(SubA^*)$. The set $AC = \{PR, SR, CR, SP\}$ is the basis for determining whether the part is suitable.

A generation algorithm is designed to generate all feasible assembly sequences. Its flowchart is shown in Fig. 6. First, the base part is determined according to assembly constraints. The base part and acquired inference results are used as the input of the algorithm. Then, the feasible parts are determined according to the logic equation E_{PR} and E_{IR}, and the judgment process is cycled until all parts are planned. The algorithm only saves the sequence of finding subsequent parts in each cycle. Finally, the connectors for all sequences are installed according to the designed procedure. The complexity of the algorithm is analyzed as follow:

Assembly A has n parts, and has m subassembly sets. The subassembly set has up to N parts. The time complexity of a single local plan is $O(N^3)$ and the time complexity of the global plan is $O(m(n/m)^3)$.

The reasoning result can also be combined with the optimization algorithm effectively and simply. On the one hand, knowledge reasoning avoids the need for algorithms to consider special sequences. On the other hand, the inference results can be easily reconstructed into the input of existing algorithms. In this paper, the ant colony algorithm is used as the optimization algorithm. Its flowchart is shown in Fig. 7. This paper makes a preliminary transformation of the original ant colony algorithm, which makes it suitable for assembly problems (Wang et al., 2005b). And the positioning relations, support relations, assembly tool, installation direction, etc. are considered by the multi-objective heuristic function (Abdullah et al., 2019).

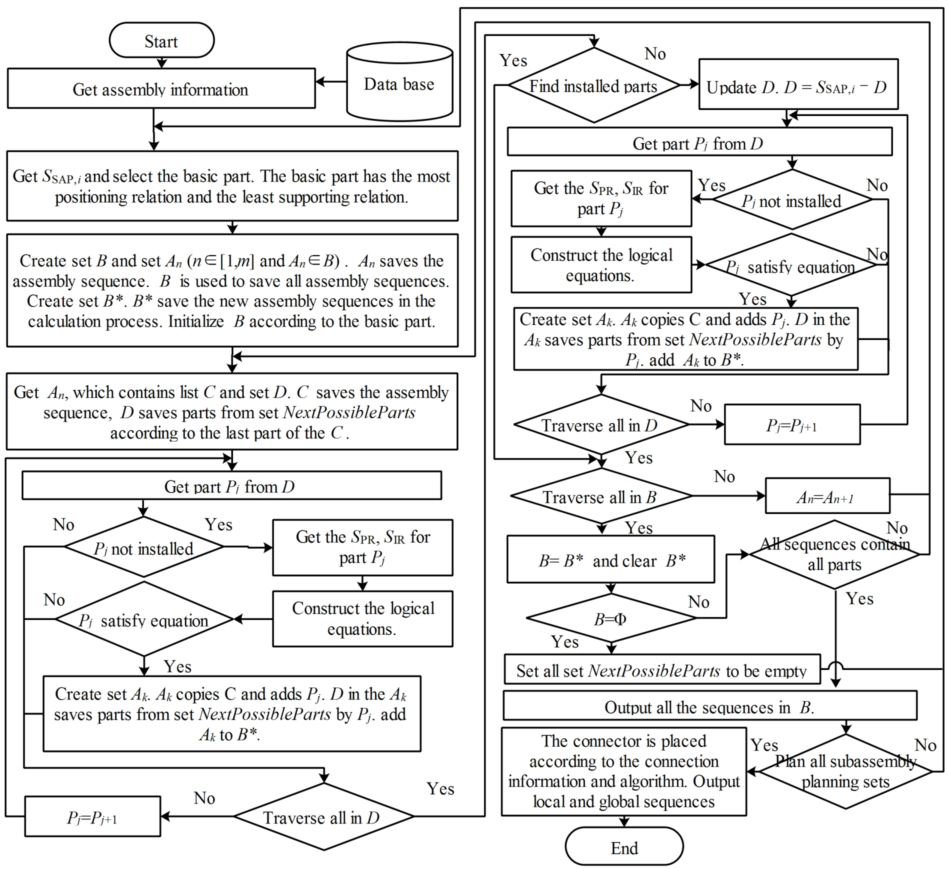


Figure 6: Flowchart of the assembly sequence generation algorithm.

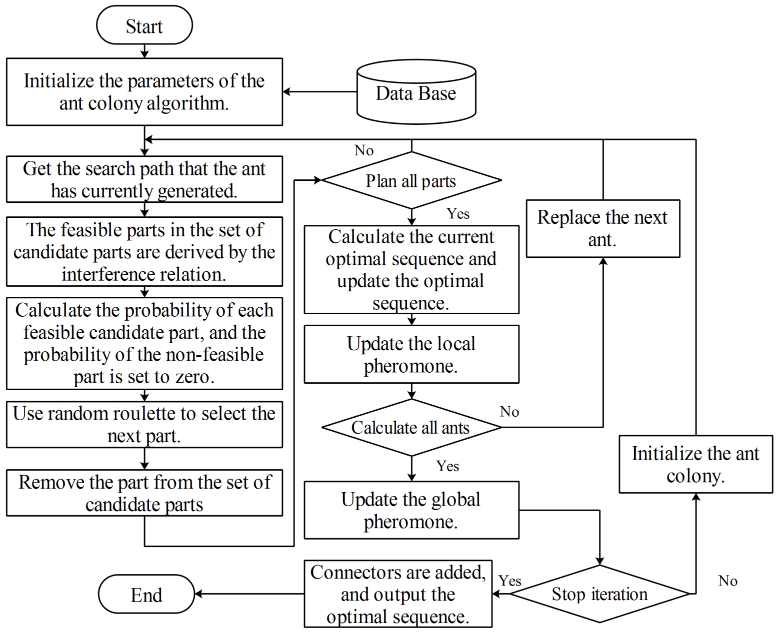


Figure 7: Flowchart of the ant colony algorithm.

5. Case study

5.1. Illustrated instance

This section uses generation of the assembly sequences of a gear reducer as an example to verify the effectiveness of the proposed ontology-based approach. The gear reducer is shown in Fig. 8. $P_1 \sim P_{16}$ indicate the main parts of the reducer, $P_{A,1} \sim P_{A,5}$ indicate auxiliary parts. $P_{C,11} \sim P_{C,52}$ indicate connectors, where $P_{C,111} \sim P_{C,163}$ are the screw, nut, and shim. According to the approach, the generation process mainly includes the following steps:

Step1 : Get assembly information. Get assembly information from design documents, CAD/PDM, and expert methods. This paper extracts the assembly tree from the CAD software's assembly file. The assembly tree of reducer contains five parent nodes, assembly $A_{S,0}$, cover $A_{S,1}$, box $A_{S,2}$, output $A_{S,3}$ and input $A_{S,4}$. Get the initial subassembly planning sets $S_{SAP,1} = \{P_{C,11}, P_{C,12}, P_{C,13}, P_{C,14}, P_{C,15}, P_{C,16}, P_{C,51}, P_{C,52}, A_{S,1}, A_{S,2}, A_{S,3}, A_{S,4}\}$;

$S_{SAP,2} = \{P_{C,21}, P_{C,22}, P_{C,23}, P_{C,24}, P_1, P_2\}$; $S_{SAP,3} = \{P_{C,31}, P_{C,32}, P_{C,33}, P_3, P_4, P_5\}$; $S_{SAP,4} = \{P_6, P_7, P_{10}, P_{11}, P_{12}, P_{13}, P_{C,4}, P_{A,1}, P_{A,4}\}$; $S_{SAP,5} = \{P_8, P_9, P_{14}, P_{15}, P_{16}, P_{A,2}, P_{A,3}, P_{A,5}\}$. The partial data of the subassembly set $S_{SAP,1}$ is shown in Table 2.

Table 2: The partial data of the subassembly set $S_{SAP,1}$.

Parts group	AR	IR_plusX	IR_minusX	IR_plusY	IR_minusY	IR_plusZ	IR_minusZ
$(A_{S,1}, A_{S,2})$	YES						YES
$(A_{S,1}, A_{S,3})$	YES	YES	YES	YES	YES		YES
$(A_{S,1}, A_{S,4})$	YES	YES	YES	YES	YES		YES
$(A_{S,2}, A_{S,3})$	YES	YES	YES	YES	YES	YES	
$(A_{S,2}, A_{S,4})$	YES	YES	YES	YES	YES	YES	
$(A_{S,3}, A_{S,4})$	YES	YES	YES		YES		
$(P_{C,51}, A_{S,1})$	YES	YES	YES	YES	YES		YES
$(P_{C,51}, A_{S,2})$	YES	YES	YES	YES	YES		YES
Connectors			CR		Connection Types		
$P_{C,111}, P_{C,121}, P_{C,131}, P_{C,141}, P_{C,151}, P_{C,161}$			$A_{S,1}, A_{S,2}$		Screw and thread connection		
$P_{C,51}, P_{C,52}$			$A_{S,1}, A_{S,2}$		Pin connection		

Step2 : Instantiate the ontology and knowledge reasoning. Firstly, the ASP ontology is instantiated according to the extracted assembly information, and then rules are executed according to the design reasoning process. Using Jena API to edit OWL files, and Prot$\acute{e}g\acute{e}$ software can perform ontologies construction, instantiation, reasoning, visualization, etc. The reasoning process and result of the example is shown in Fig. 9. Subassembly planning sets:$S_{SAP,1} = \{P_{C,1\_1}, P_{C,1\_2}, P_{C,5}, A_{S,1}, A_{S,2}, A_{S,3}, A_{S,4}\}$; $S_{SAP,2} = \{P_{C,2}, P_1, P_2\}$; $S_{SAP,3} = \{P_{C,3}, P_3, P_4, P_5\}$; $S_{SAP,4} = \{P_6, P_7, P_{10}, P_{11}, A_{S,5}, P_{A,1}, P_{A,4}\}$, $A_{S,5} = \{P_{12}, P_{13}, P_{C,4}\}$; $S_{SAP,5} = \{P_8, P_9, P_{14}, P_{15}, P_{16}, P_{A,2}, P_{A,3}, P_{A,5}\}$.

Step3 : Generate assembly sequences. Execute the algorithm in the Java development environment. For example, $A_{S,1}, A_{S,2}, A_{S,3}, A_{S,4}$ are sorted by the algorithm, then connectors and special sequences are added to get the entire sequence. The generated assembly sequences and planning program are shown in Table 3 and Fig. 10. Two of feasible sequences and the optimal sequence are listed in the Table 3.

5.2. Comparative study

a) Compared with the geometric feature methods: This paper also uses the geometric features of product as the judgment basis for the generation algorithm.

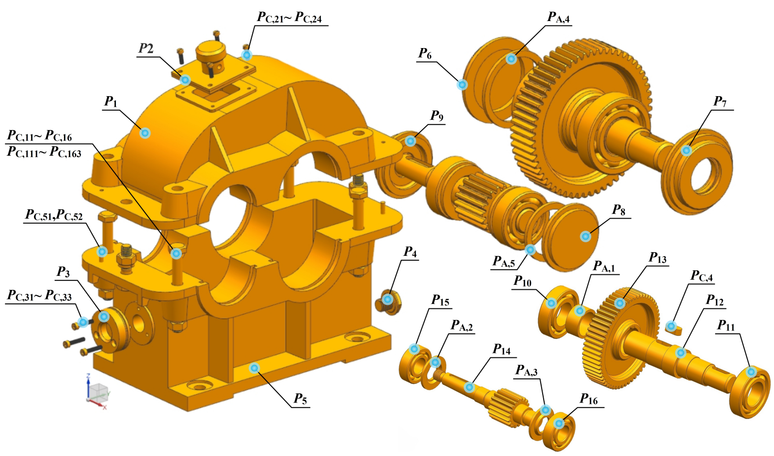


Figure 8: A gear reducer used to verify the effectiveness of the approach.

Table 3: Assembly sequences generated by the ontology-based approach.

Subassembly	Total	Local assembly sequence
$S_{SAP,1}$	2	$A_{S,2} \to A_{S,4} \to A_{S,3} \to A_{S,1}$ $A_{S,2} \to A_{S,3} \to A_{S,4} \to A_{S,1}$
$S_{SAP,2}$	1	$P_1 \to P_2$
$S_{SAP,3}$	2	$P_5 \to P_4 \to P_3$ $P_5 \to P_3 \to P_4$
$S_{SAP,4}$	2	$A_{S,5} \to P_{A,1} \to P_{10} \to P_{A,4} \to P_6 \to P_{11} \to P_7$ $A_{S,5} \to P_{11} \to P_7 \to P_{A,1} \to P_{10} \to P_{A,4} \to P_6$
$S_{SAP,5}$	2	$P_{14} \to P_{A,2} \to P_{15} \to P_9 \to P_{A,3} \to P_{16} \to P_{A,5} \to P_8$ $P_{14} \to P_{A,3} \to P_{16} \to P_{A,5} \to P_8 \to P_{A,2} \to P_{15} \to P_9$
ID	Global assembly sequence	
1	$P_5 \to P_4 \to P_3 \to (P_{C,31}, P_{C,32}, P_{C,33}) \to P_{12} \to P_{C,4} \to P_{13} \to P_{A,1} \to P_{10} \to P_{A,4} \to P_6 \to P_{11} \to P_7 \to P_{14} \to P_{A,2} \to P_{15} \to P_9 \to P_{A,3} \to P_{16} \to P_{A,5} \to P_8 \to P_1 \to P_2 \to (P_{C,21}, P_{C,22}, P_{C,23}, P_{C,24}) \to (P_{C,111} \to P_{C,113} \to P_{C,112}, P_{C,121} \to P_{C,123} \to P_{C,122}, P_{C,131} \to P_{C,133} \to P_{C,132}, P_{C,141} \to P_{C,143} \to P_{C,142}) \to (P_{C,151} \to P_{C,153} \to P_{C,152}, P_{C,161} \to P_{C,163} \to P_{C,162}) \to (P_{C,51}, P_{C,52})$	
2	$P_5 \to P_4 \to P_3 \to (P_{C,31}, P_{C,32}, P_{C,33}) \to P_{12} \to P_{C,4} \to P_{13} \to P_{11} \to P_7 \to P_{A,1} \to P_{10} \to P_{A,4} \to P_6 \to P_{14} \to P_{A,3} \to P_{16} \to P_{A,5} \to P_8 \to P_{A,2} \to P_{15} \to P_9 \to P_1 \to P_2 \to (P_{C,21}, P_{C,22}, P_{C,23}, P_{C,24}) \to (P_{C,111} \to P_{C,113} \to P_{C,112}, P_{C,121} \to P_{C,123} \to P_{C,122}, P_{C,131} \to P_{C,133} \to P_{C,132}, P_{C,141} \to P_{C,143} \to P_{C,142}) \to (P_{C,151} \to P_{C,153} \to P_{C,152}, P_{C,161} \to P_{C,163} \to P_{C,162}) \to (P_{C,51}, P_{C,52})$	
	Optimal assembly sequence	
	$P_5 \to P_4 \to P_3 \to (P_{C,31}, P_{C,32}, P_{C,33}) \to P_{12} \to P_{C,4} \to P_{13} \to P_{A,1} \to P_{10} \to P_{A,4} \to P_6 \to P_{11} \to P_7 \to P_{14} \to P_{A,3} \to P_{16} \to P_{A,5} \to P_8 \to P_{A,2} \to P_{15} \to P_9 \to P_1 \to P_2 \to (P_{C,21}, P_{C,22}, P_{C,23}, P_{C,24}) \to (P_{C,111} \to P_{C,113} \to P_{C,112}, P_{C,121} \to P_{C,123} \to P_{C,122}, P_{C,131} \to P_{C,133} \to P_{C,132}, P_{C,141} \to P_{C,143} \to P_{C,142}) \to (P_{C,151} \to P_{C,153} \to P_{C,152}, P_{C,161} \to P_{C,163} \to P_{C,162}) \to (P_{C,51}, P_{C,52})$	

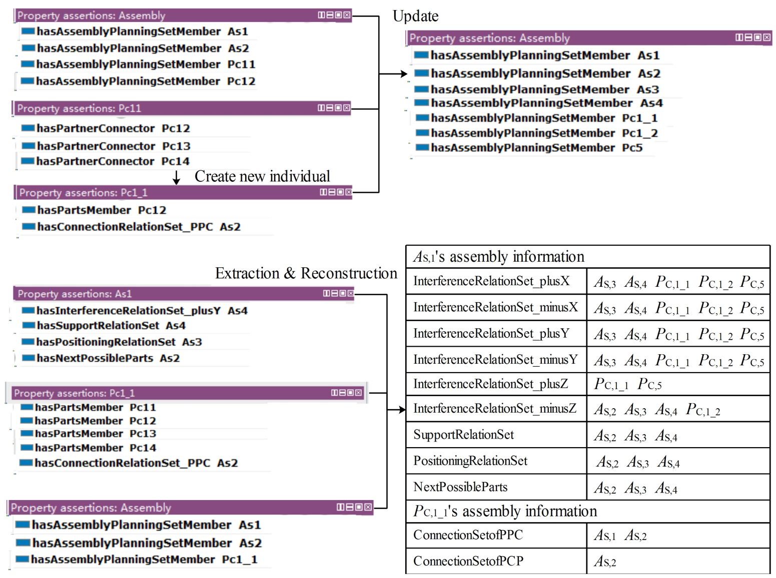
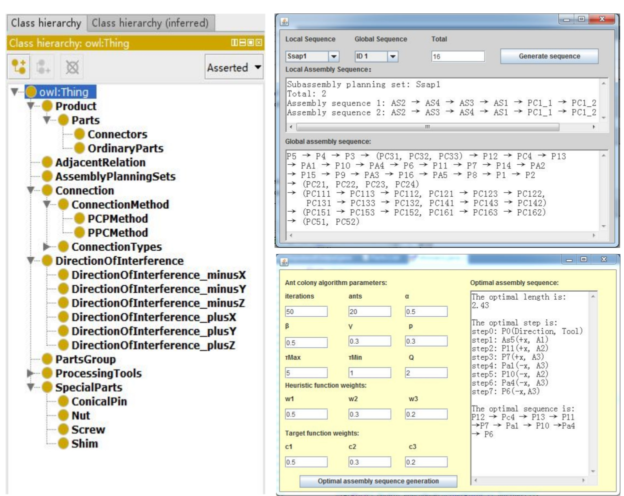


Figure 9: Extraction and reasoning of the information of the gear reducer.



Notes: Number of iterations N; Number of ants M; Pheromone weight α; Transition probability weight β; Global volatility γ; Local volatility p; Maximum pheromone τMax; Minimum pheromone τMin; Adjustment parameter Q; Redirection weight W_1; Stability weight W_2; Tool weight W_3; Redirection weight C_1; Stability weight C_2; Tool weight C_3.

Figure 10: Assembly information in the ontology and assembly sequences generated in the system.

However, since assembly experience and connection semantics are utilized, the presented ontology method narrows the scope of feasible sequences and filters sequences that do not conform to assembly habits. With the existence of part groups, connectors, and special sequences, the number of parts that the actual algorithm needs to calculate will be reduced accordingly.

The polychromatic sets theory (Zhao and Li, 2009) was used to generate the sequences of a pump. The assembly contains 10 parts and produces 40 suitable sequences. In the presented ontology method, there are 4 parts that need to be sorted according to the inference results, and the algorithm produces 2 suitable sequences. The reason for the significant reduction in reasonable sequences is the special sequence of shaft and the installation of connectors. The assembly is shown in Figure 11, and the results are shown in Table 4.

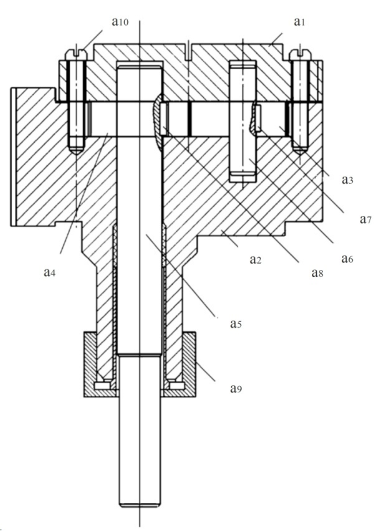


Figure 11: Assembly body profile of a pump (Zhao and Li, 2009).

A vise is used to validate the concurrent method based on directed constraint graph (Hu et al., 2010), which has 15 parts and produces a disassembly scheme (which can generate 720 assembly sequences). Finally, a parallel assembly directed graph is generated by parallelization strategy and disassembly scheme (which can generate 8 assembly sequences). In the presented ontology method,

Table 4: Assembly sequences generated by the ontology-based approach.

Subassembly	Local assembly sequence
$S_{SAP,1}$	$a_2 \rightarrow A_{s,1} \rightarrow A_{s,2} \rightarrow a_1; a_2 \rightarrow A_{s,2} \rightarrow A_{s,1} \rightarrow a_1$
$A_{s,1}$	$a_5 \rightarrow a_8 \rightarrow a_4$
$A_{s,2}$	$a_6 \rightarrow a_7 \rightarrow a_3$
ID	Global assembly sequence
1	$a_2 \rightarrow a_5 \rightarrow a_8 \rightarrow a_4 \rightarrow a_9 \rightarrow a_6 \rightarrow a_7 \rightarrow a_3 \rightarrow a_1 \rightarrow a_{10}$
2	$a_2 \rightarrow a_6 \rightarrow a_7 \rightarrow a_3 \rightarrow a_5 \rightarrow a_8 \rightarrow a_4 \rightarrow a_9 \rightarrow a_1 \rightarrow a_{10}$

there are 6 parts that need to be sorted, and the algorithm produces 8 suitable sequences. The sequences have a similar effect to the current method. Since the parallel connectors are sorted in the process of reasoning. The assembly is shown in Figure 12. Four of feasible local sequences and two of global sequences are shown in Table 5.

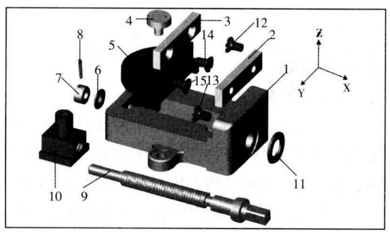


Figure 12: The assembly exploded diagram of a vise (Hu et al., 2010).

Table 5: Assembly sequences generated by the ontology-based approach.

Subassembly	Local assembly sequence
$S_{SAP,1}$	$1 \rightarrow 2 \rightarrow 10 \rightarrow 9 \rightarrow 5 \rightarrow 3; 1 \rightarrow 10 \rightarrow 9 \rightarrow 5 \rightarrow 3 \rightarrow 2$ $1 \rightarrow 5 \rightarrow 10 \rightarrow 9 \rightarrow 2 \rightarrow 3; 1 \rightarrow 10 \rightarrow 5 \rightarrow 3 \rightarrow 9 \rightarrow 2$
ID	Global assembly sequence
1	$1 \rightarrow 2 \rightarrow (12, 13) \rightarrow 10 \rightarrow 11 \rightarrow 9 \rightarrow (6 \rightarrow 7 \rightarrow 8) \rightarrow 5 \rightarrow 4$ $\rightarrow 3 \rightarrow (14, 15)$
2	$1 \rightarrow 10 \rightarrow 11 \rightarrow 9 \rightarrow (6 \rightarrow 7 \rightarrow 8) \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow (14, 15)$ $\rightarrow 2 \rightarrow (12, 13)$

b) Compared with the knowledge-based methods: The presented ontology method is one of the knowledge-based methods and has an advantage in semantic representation. Qiao et al. (2018) and Jiang et al. (2018) use ontology to describe assembly knowledge. Based on assembly knowledge, a feasible assembly sequence is generated by rules, inference units and algorithms. Starting from the requirements of automatic generation algorithms, this paper constructs rules and reasoning processes with the goal of reducing the computational complexity of products, narrowing the range of reasonable sequences, and generating optimal sequences that conform to assembly habits. Therefore, this paper differs from other ontology-based methods in rules and inference processes, and is more suitable for sequence optimization and automatic generation algorithms.

A transmission is instance of Qiao et al. (2018), which contains 22 parts and produces a reasonable sequence through rule reasoning. The assembly is shown in Figure 13. In our method, there are 8 parts that need to be sorted according to the inference results, and the algorithm produces 18 suitable sequences. Unlike instance of subsection 5.1, which is handled as a part group, logical equations are used to calculate the priority relations. If P_2, $P_{c,1}$ and P_8 are taken as a whole, they interfere with P_4. Therefore, they cannot be combined into a new part group. Simultaneously, interference checks are performed on $P_{c,2}$, P_8 and P_7 according to the designed inference steps. Get a priority relation that P_8 must be installed before P_7. Hence, the logic equation of P_8 is $E_{IR} = (\bigwedge S_{IR}i) \bigvee P_7 \bigvee P_2$. Four of feasible local sequences and two of global sequences are shown in Table 6.

Table 6: Assembly sequences generated by the ontology-based approach.

Subassembly	Local assembly sequence
$S_{SAP,1}$	$P_8 \to P_5 \to P_6 \to P_1 \to P_7 \to P_4 \to P_3 \to P_2$; $P_8 \to P_5 \to P_6 \to P_1 \to P_4 \to P_3 \to P_7 \to P_2$ $P_8 \to P_4 \to P_3 \to P_1 \to P_5 \to P_6 \to P_2 \to P_7$; $P_8 \to P_4 \to P_3 \to P_1 \to P_2 \to P_5 \to P_6 \to P_7$
ID	**Global assembly sequence**
1	$P_8 \to P_5 \to P_6 \to P_1 \to (P_{C,9}, P_{C,10}, P_{C,11}, P_{C,12}, P_{C,13}, P_{C,14}) \to P_{C,2} \to P_7 \to P_4$ $\to P_3 \to (P_{C,3}, P_{C,4}, P_{C,5}, P_{C,6}, P_{C,7}, P_{C,8}) \to P_{C,1} \to P_2$
2	$P_8 \to P_4 \to P_1 \to P_3 \to (P_{C,3}, P_{C,4}, P_{C,5}, P_{C,6}, P_{C,7}, P_{C,8}) \to P_{C,1} \to P_2 \to P_5 \to P_6$ $\to (P_{C,9}, P_{C,10}, P_{C,11}, P_{C,12}, P_{C,13}, P_{C,14}) \to P_{C,2} \to P_7$

A centrifugal pump is taken as an instance of Jiang et al. (2018), which contains 26 parts and produces a disassembly scheme (which can be sorted into 1356 assembly sequences) through the design of algorithm. This paper generates 112 assembly sequences. The reason for the sequence reduction is that the algorithm considers the principle of proximity installation and uses the relative position information implied by the adjacency relation. The assembly is shown in Figure

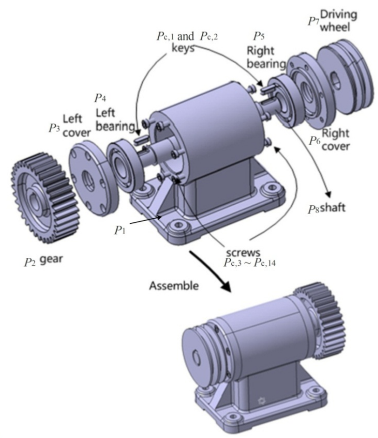


Figure 13: Structure of the transmission (Qiao et al., 2018).

14. $S_{SAP,1} = \{1, 2, 4, 7, 8, 9, 11, 12, 13, 14, 15, 16, 18, 21, 26, A_{s,1}, A_{s,2}, A_{s,3}, A_{s,4}\}$, $A_{s,1} = \{6, 19\}, A_{s,2} = \{17, 25\}, A_{s,3} = \{3, 10, 20, 22\}, A_{s,4} = \{5, 23, 24\}$. Like Jiang's processing, we used $A_{s,1} \sim A_{s,4}$ as subassemblies for sequence planning. According to the disassembly experience, the part 17 is artificially set as a base part. Two of feasible local sequences and two of global sequences are shown in Table 7.

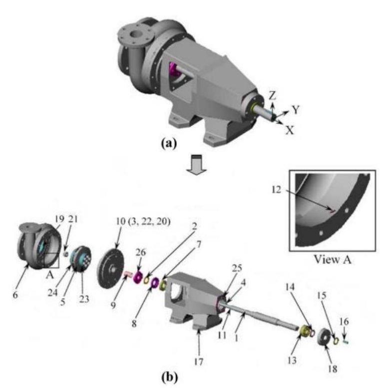


Figure 14: The exploded diagram of a centrifugal pump (Jiang et al., 2018).

c) Compared with the heuristic algorithms: The heuristic algorithm enables simultaneous sequence generation and quality assessment. However, the unified

Table 7: Assembly sequences generated by the ontology-based approach.

Subassembly	Local assembly sequence
$S_{SAP,1}$	$A_{s,2} \rightarrow 7 \rightarrow 1 \rightarrow 13 \rightarrow 14 \rightarrow 18 \rightarrow 15 \rightarrow 8 \rightarrow 2 \rightarrow 26 \rightarrow A_{s,3} \rightarrow 9 \rightarrow 11 \rightarrow 4 \rightarrow A_{s,4} \rightarrow 12 \rightarrow 21 \rightarrow A_{s,1} \rightarrow 16$
	$A_{s,2} \rightarrow 13 \rightarrow 1 \rightarrow 7 \rightarrow 8 \rightarrow 2 \rightarrow 26 \rightarrow A_{s,3} \rightarrow 9 \rightarrow 11 \rightarrow 4 \rightarrow A_{s,4} \rightarrow 12 \rightarrow 21 \rightarrow A_{s,1} \rightarrow 14 \rightarrow 18 \rightarrow 15 \rightarrow 16$
ID	**Global assembly sequence**
1	$A_{s,2} \rightarrow 7 \rightarrow 1 \rightarrow 13 \rightarrow 14 \rightarrow 18 \rightarrow bolt \rightarrow 15 \rightarrow 8 \rightarrow 2 \rightarrow 26 \rightarrow A_{s,3} \rightarrow (double\_screw\_bolt\_and\_nut\_3) \rightarrow (double\_screw\_bolt\_and\_nut\_2) \rightarrow 9 \rightarrow 11 \rightarrow 4 \rightarrow A_{s,4} \rightarrow 12 \rightarrow 21 \rightarrow pin \rightarrow A_{s,1} \rightarrow (double\_screw\_bolt\_and\_nut\_1) \rightarrow 16$
2	$A_{s,2} \rightarrow 13 \rightarrow 1 \rightarrow 7 \rightarrow 8 \rightarrow 2 \rightarrow 26 \rightarrow A_{s,3} \rightarrow (double\_screw\_bolt\_and\_nut\_3) \rightarrow (double\_screw\_bolt\_and\_nut\_2) \rightarrow 9 \rightarrow 11 \rightarrow 4 \rightarrow A_{s,4} \rightarrow 12 \rightarrow 21 \rightarrow pin \rightarrow A_{s,1} \rightarrow (double\_screw\_bolt\_and\_nut\_1) \rightarrow 14 \rightarrow 18 \rightarrow bolt \rightarrow 15 \rightarrow 16$

evaluation functions used in heuristic algorithms are difficult to incorporate into individual assembly experience. The presented ontology method controls the special sequences in the form of part groups and interference logic equations. The part groups are calculated as a normal part, and the interference relation is a condition that must be satisfied for each sequence. And then the connectors are independently planned. Therefore, using the presented ontology method, assembly experience can be easily integrated into existing heuristic algorithms without special modifications.

Yu and Wang (2013) used an improved algorithm based on Ant Colony optimizer, and the parallelism of parts was integrated into the algorithm through calculation. The assembly is shown in Figure 15. The optimal sequence has priority to install parallel parts, but there are cases where the nut is installed before the screw and the nut and screw are not installed continuously. The presented ontology method has 8 parts in the algorithm after reasoning. In terms of optimization, common installation directions, installation tools and stability are considered as evaluation factors. The function (1), function (2) and function (3) define the heuristic function of installation direction, stability, and installation tools, respectively.

$$d_{ij} = \begin{cases} 0.2, & d_i \neq d_j \\ 1, & d_i = d_j \end{cases} \tag{1}$$

Where d_i represents the installation direction of part i, and d_{ij} represents the redirection information from part i to part j.

$$s_{ij} = \begin{cases} 0.1 & E_{PRj} = 0 \text{ And } E_{SRj} = 0 \\ 0.5 & E_{PRj} = 1 \text{ Or } E_{SRj} = 1 \\ 0.8 & E_{PRj} = 1 \text{ And } E_{SRj} = 1 \\ 1 & E_{PRj} = 1 \text{ And } E_{SRj} = 1 \text{ And } i \in S_{PRj} \end{cases} \tag{2}$$

Where S_{ij} represents stability information from part i to part j.

$$t_{ij} = \begin{cases} 0.2, & t_i \neq t_j \\ 1, & t_i = t_j \end{cases} \tag{3}$$

Where t_i represents the installation tool of part i, and t_{ij} represents replacement information of tool from part i to part j. An objective function is defined as shown in function (4).

$$f_{ij} = w_1 d_{ij} + w_2 s_{ij} + w_3 t_{ij} \tag{4}$$

Where f_{ij} represents the multi-objective heuristic function, $w_1 \sim w_3$ represent weights, and w_1+w_3+w_3=1.

Similarly, an evaluation function of sequence is constructed as shown in function (5).

$$S = w_1 D + w_2 S + w_3 T \tag{5}$$

Where $w_1 \sim w_3$ are weights, and w_1+w_3+w_3=1. D represents the number of times the installation direction has changed. T represents the number of times the tool has changed. S represents the stability of the sequence as shown in function (6) and (7).

$$S = \sum_{k=1}^{n} S_{ij} \tag{6}$$

$$s_{ij} = \begin{cases} 1 & E_{PRj} = 0 \text{ And } E_{SRj} = 0 \\ 0.8 & E_{PRj} = 1 \text{ Or } E_{SRj} = 1 \\ 0.5 & E_{PRj} = 1 \text{ And } E_{SRj} = 1 \\ 0.1 & E_{PRj} = 1 \text{ And } E_{SRj} = 1 \text{ And } i \in S_{PRj} \end{cases} \tag{7}$$

The optimal sequence is shown in Table 8. The sequence changes once in the installation direction and 3 times in the installation tools.

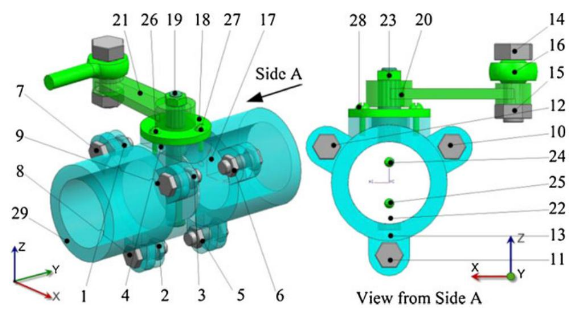


Figure 15: Example of valve assembly (Yu and Wang, 2013).

Table 8: Optimal assembly sequences generated by the ontology-based approach.

Optimal sequence	$17(+z, T_1) \rightarrow 29(+z, T_1) \rightarrow 19(+z, T_3) \rightarrow 18(+z, T_1) \rightarrow 21(+z, T_1)$ $\rightarrow 16(+z, T_1) \rightarrow 13(+z, T_1) \rightarrow 22(+y, T_2)$
Global assembly sequence	$17 \rightarrow 29 \rightarrow (7 \rightarrow 1, 8 \rightarrow 2, 9 \rightarrow 3) \rightarrow 19 \rightarrow 18 \rightarrow (26, 27, 28) \rightarrow 20$ $\rightarrow 21 \rightarrow 23 \rightarrow 16 \rightarrow (14 \rightarrow 15) \rightarrow 13 \rightarrow (10 \rightarrow 4, 11 \rightarrow 5, 12 \rightarrow 6)$ $\rightarrow 22 \rightarrow (24, 25)$

Rashid (2017) used a hybrid algorithm based on Ant Colony and Grey Wolf optimizers, and the question-answer and precedence graph were used for the acquisition and presentation of assembly experience. The use of precedence graph and matrix (Wei et al., 2014) incorporates assembly experience, but cannot be directly used for planning new assemblies. Ontology has a natural advantage in the accumulation of experience, sharing, and reuse, and can store and reason the priority relations as rules of different granularity. The table vise of Rashid contains 21 parts, and the presented ontology method has 8 parts in the algorithm after reasoning. The task-based precedence graph is converted to part-based ontology rules (without connectors) and priority relations are controlled by part groups and logic equations. For example, assume that the model number of P_5 is entity 5 and the model number of P_{21} is entity 21. Use the following rule to describe the assembly rules for parts that are limited to a specific model.

$Parts(?x) \wedge Parts(?y) \wedge Base(?m) \wedge Plate(?n) \wedge isTypeOfParts(?x,?m)$
$\wedge isTypeOfParts(?x,?n) \wedge sameAs(?m,21) \wedge sameAs(?n,5) - >$
$hasInstallBefore(?x,?y) \wedge hasInstallAfter(?y,?x)$

The assembly is shown in Figure 16. $S_{SAP,1} = \{A_{s,1}, P_2, A_{s,2}, P_4, P_5, P_6, P_{12}, P_{13}, P_{14}, P_{15}, P_{17}, P_{18}, P_{19}, P_{21}, P_{C,1}, P_{C,2}\}, P_{C,1} = \{P_{10}, P_{11}\}, P_{C,2} = \{P_{16}, P_{20}\}, A_{s,1} = \{P_1, P_8, P_9\}, A_{s,2} = \{P_3, P_7\}$. The optimal sequence is shown in Table 9. The sequence changes once in the installation direction and 2 times in the installation tools.

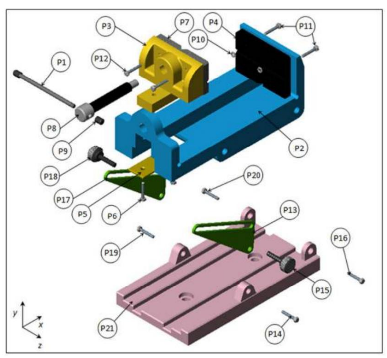


Figure 16: Table vise assembly (Rashid, 2017).

Table 9: Optimal assembly sequences generated by the ontology-based approach.

Subassembly	Optimal sequence/Local assembly sequence
$S_{SAP,1}$	$P_2(+y,T_1) \rightarrow P_{17}(+y,T_1) \rightarrow P_{13}(+y,T_1) \rightarrow P_4(+y,T_1) \rightarrow A_{s,2}(+y,T_1)$ $\rightarrow A_{s,1}(-x,T_2) \rightarrow P_5(-x,T_1) \rightarrow P_{21}(-x,T_1)$
$A_{s,1}$	$P_8 \rightarrow P_1 \rightarrow P_9$
$A_{s,2}$	$P_7 \rightarrow P_3$
Global assembly sequence	$P_2 \rightarrow P_{17} \rightarrow P_{18} \rightarrow P_{13} \rightarrow P_{15} \rightarrow P_4 \rightarrow (P_{11} \rightarrow P_{10}) \rightarrow (P_7 \rightarrow P_3) \rightarrow P_{12}$ $\rightarrow (P_8 \rightarrow P_1 \rightarrow P_9) \rightarrow P_5 \rightarrow P_6 \rightarrow P_{21} \rightarrow P_{19} \rightarrow P_{14} \rightarrow (P_{16}, P_{20})$

A summary of the comparison results of the above six methods and the presented ontology method is provided in Table 10. As can be seen from the table, the proposed method can further reduce the number of solutions and remove sequences that do not meet the actual assembly environment and assembly habits. The proposed method can effectively reduce the calculation space, consider the priority relations, and ensure the parallelism of the installation. In addition, it is more convenient to transplant, because the algorithm only needs to consider more common assembly factors and the efficiency of the algorithm itself.

Table 10: Comparison results of the proposed ontology-based approach and other methods.

Assembly	PN	Planning method	AN	SN
Gear reducer	49	O	24	16, OS
Pump I	10	Zhao and Li (2009) $\parallel O$	10 $\parallel$ 4	40 $\parallel$ 2
Vise I	15	Hu et al. (2010) $\parallel O$	15 $\parallel$ 6	8 $\parallel$ 8
Transmission	22	Qiao et al. (2018) $\parallel O$	T(10) $\parallel$ 8	1 $\parallel$ 18
Pump II	26	Jiang et al. (2018) $\parallel O$	19 $\parallel$ 19	1356 $\parallel$ 112
Valve	29	Yu and Wang (2013) $\parallel O$	29 $\parallel$ 8	$OS \parallel OS$
Vise II	21	Rashid (2017) $\parallel O$	T(19) $\parallel$ 8	$OS \parallel OS$

Notes: PN represents the number of parts. AN represents the number of parts in the algorithm or inference engine. SN represents the number of assembly sequences. O represents the presented ontology method. OS represents an optimal sequence. $T(x)$ represents the assembly operations, and x the number of assembly operations.

6. Conclusion

This paper established an assembly model and designed an ontology-based assembly sequence automatic generation approach on the basis of this model. The working process of the approach was demonstrated via an engineering example, and the effectiveness of the approach in reducing unreasonable assembly sequences and amount of calculation was verified. Compared with other methods, the approach has the following aspects of advantages:

(1) Assembly knowledge representation and automatic reasoning. Based on the assembly ontology, assembly knowledge can be interpreted by computers, and combined rules can realize automatic reasoning of assembly knowledge. The computation amount of the algorithm is reduced, and the assembly sequence is guaranteed to conform to the actual assembly habit.

(2) High flexibility and scalability. Ontology has certain advantages for the reconstruction and expansion of knowledge, and it is easy to add new knowl-

edge. The design requirements of the algorithm are simplified and more easily combined with existing algorithms.

(3) Knowledge integration and sharing. ASP is a comprehensive planning task based on assembly knowledge. Ontology has natural advantages in integration and sharing of knowledge.

Further research can proceed from the following aspects:

(1) Automatic construction of ontology. This ontology needs to be further enriched to accommodate more complex products. Automatic recognition of concepts and entities in databases, web pages and texts would be achieved through R2RML, machine learning methods, etc.

(2) Automatic generation of rules. By assembling sequence cases and decision trees, the feature probability and decision process of the special sequence would be calculated, and finally automatically converted into rules.

7. Acknowledgements

The authors would like to acknowledge the financial support by the National Natural Science Foundation of China (61562016 and 51765012), and Guangxi Graduate Education Innovation Program (No.YCSZ2015148).

8. References

Abdullah, M.A., Rashid, M.F.F.A., Ghazalli, Z., 2019. Optimization of assembly sequence planning using soft computing approaches: a review. Archives of Computational Methods in Engineering 26, 461–474.

Baldwin, D.F., Abell, T.E., Lui, M.C., De Fazio, T.L., Whitney, D.E., 1991. An integrated computer aid for generating and evaluating assembly sequences for mechanical products. IEEE Transactions on Robotics and Automation 7, 78–94.

Barbau, R., Krima, S., Rachuri, S., Narayanan, A., Fiorentini, X., Foufou, S., Sriram, R.D., 2012. Ontostep: Enriching product model data using ontologies. Computer-Aided Design 44, 575–590.

Stanford Center for Biomedical Informatics Research, S.U., . Protege 5.0.0. `http://protege.stanford.edu/`. Accessed 1 June, 2018.

Bourjault, A., 1984. Contribution a une Approche Methodologique de L'Assemblage Automatise: Elaboration Automatique des Sequences Operatoires, "Contribution to the methodology of automated assembly: automatic generation of operations sequences". Ph.D. thesis. Thése d'Etat, Université de Franche-Comté.

Chen, J., Zhang, S., Li, X., Chen, K., 2016a. Identifying and generating subassemblies in disassembly sequence planning. Chinese Journal of Engineering Design 23, 1–7.

Chen, S., Yi, J., Jiang, H., Zhu, X., 2016b. Ontology and cbr based automated decision-making method for the disassembly of mechanical products. Advanced Engineering Informatics 30, 564–584.

Chen, X., Gao, S., Yang, Y., Zhang, S., 2012. Multi-level assembly model for top-down design of mechanical products. Computer-Aided Design 44, 1033–1048.

Cheng, Y., Yi, J., Jiang, H., Lin, Y., Chen, S., 2012. A computer disassembly method based on ontology-case-based reasoning, in: Advanced Materials Research, Trans Tech Publ. pp. 1398–1402.

Cochrane, S., Young, R., Case, K., Harding, J., Gao, J., Dani, S., Baxter, D., 2009. Manufacturing knowledge verification in design support systems. International Journal of Production Research 47, 3179–3204.

Dong, T., Tong, R., Zhang, L., Dong, J., 2007. A knowledge-based approach to assembly sequence planning. The International Journal of Advanced Manufacturing Technology 32, 1232–1244.

Fensel, D.A., 2003. Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer.

Fernández-López, M., Gómez-Pérez, A., 2002. Overview and analysis of methodologies for building ontologies. The Knowledge Engineering Review 17, 129–156.

Gao, J., Mou, P., Xiang, D., Duan, G., Wang, J., 2004. Unigraphics-based automatic acquirement of part adjacency matrix. China Mechanical Engineering 15, 49–51.

Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition 5, 199–220.

Gruhier, E., Demoly, F., Dutartre, O., Abboudi, S., Gomes, S., 2015. A formal ontology-based spatiotemporal mereotopology for integrated product design and assembly sequence planning. Advanced Engineering Informatics 29, 495–512.

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M., . Swrl: A semantic web rule language combining owl and ruleml. `http://www.w3.org/Submission/SWRL/`. Accessed 1 June, 2018.

Hu, X., Zhu, W., Yu, T., 2010. Concurrent method of assembly sequence based on directed constraint graph. Machinery Design and Manufacture 4, 163–165.

Jiang, H., Yi, J., Zhu, X., Li, Z., 2018. Generating disassembly tasks for selective disassembly using ontology-based disassembly knowledge representation. Assembly Automation 38, 113–124.

Kim, J., Kim, K., Lee, J., Jeong, J., 2005. Generation of assembly models from kinematic constraints. The International Journal of Advanced Manufacturing Technology 26, 131–137.

Kim, K.Y., Manley, D.G., Yang, H., 2006. Ontology-based assembly design and information sharing for collaborative product development. Computer-Aided Design 38, 1233–1250.

Lu, C., Yang, Z., 2016. Integrated assembly sequence planning and assembly line balancing with ant colony optimization approach. The International Journal of Advanced Manufacturing Technology 83, 243–256.

Lv, H., Lu, C., Zha, J., 2010. A hybrid dpso-sa approach to assembly sequence planning, in: 2010 IEEE International Conference on Mechatronics and Automation, IEEE. pp. 1998–2003.

McGuinness, D.L., Harmelen, F.v., . Owl web ontology language overview. `https://www.w3.org/TR/owl-features/`. Accessed 1 June, 2018.

Meng, Y., Gu, T., Chang, L., Li, F., 2016. Assembly ontology for assembly sequence planning. Pattern Recognition and Artificial Intelligence 29, 203–215.

Motik, B., Sattler, U., Studer, R., 2005. Query answering for owl-dl with rules. Web Semantics: Science, Services and Agents on the World Wide Web 3, 41–60.

Ou, L., Xu, X., 2013. Relationship matrix based automatic assembly sequence generation from a cad model. Computer-Aided Design 45, 1053–1067.

OWL Working Group, W., . Owl 2 web ontology language document overview (second edition). `https://www.w3.org/TR/owl2-overview/`. Accessed 3 June, 2018.

Qiao, L., Qie, Y., Zhu, Z., Zhu, Y., uz Zaman, U.K., Anwer, N., 2018. An ontology-based modelling and reasoning framework for assembly sequence planning. The International Journal of Advanced Manufacturing Technology 94, 4187–4197.

Rashid, M.F.F.A., 2017. A hybrid ant-wolf algorithm to optimize assembly sequence planning problem. Assembly Automation 37, 238–248.

Staab, S., Studer, R., 2010. Handbook on ontologies. Springer Science & Business Media.

Swaminathan, A., Shaikh, S.A., Barber, K.S., 1998. Design of an experience-based assembly sequence planner for mechanical assemblies. Robotica 16, 265–283.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., Tomiyama, T., 1996. Supporting conceptual design based on the function-behavior-state modeler. Ai Edam 10, 275–288.

Wang, J., Li, S., Liu, J., Zhong, Y., 2005a. Computer aided assembly planning: a survey. Journal of Engineering Graphics 2.

Wang, J., Liu, J., Zhong, Y., 2005b. A novel ant colony algorithm for assembly sequence planning. The International Journal of Advanced Manufacturing Technology 25, 1137–1143.

Wei, W., Guo, C., Duan, X., Liu, Y., 2014. Method of assembly sequence planning based on ant colony genetic hybrid algorithm. Journal of System

Simulation 26, 1684–1691.

Xu, X., Wang, L., Newman, S.T., 2011. Computer-aided process planning–a critical review of recent developments and future trends. International Journal of Computer Integrated Manufacturing 24, 1–31.

Yin, Z., Ding, H., Li, H., Xiong, Y., 2003. A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. Computer-Aided Design 35, 37–56.

Yu, J., Wang, C., 2013. A max–min ant colony system for assembly sequence planning. The International Journal of Advanced Manufacturing Technology 67, 2819–2835.

Yu, J., Wang, C., Zhang, W., 2010. Automatic acquiring method for assembly relation matrix of complex product. Computer Integrated Manufacturing Systems 2, 249–255.

Zhang, W., Qu, R., Xu, M., Luo, X., 2016. Method for automatic generation of assembly interference matrix of complex products. Journal of Mechanical Engineering 52, 139–148.

Zhang, W., Tor, S., Britton, G., 2005. A graph and matrix representation scheme for functional design of mechanical products. The International Journal of Advanced Manufacturing Technology 25, 221–232.

Zhao, S., Li, Z., 2009. Formalized reasoning method for assembly sequences based on polychromatic sets theory. The International Journal of Advanced Manufacturing Technology 42, 993–1004.

Zhu, B., Roy, U., 2018. Modeling and validation of a web ontology language based disassembly planning information model. Journal of Computing and Information Science in Engineering 18, 021015.

Zhu, L., Jayaram, U., Kim, O., 2012. Semantic applications enabling reasoning in product assembly ontologiesmoving past modeling. Journal of Computing and Information Science in Engineering 12, 011009.